

(12) **UK Patent Application** (19) **GB** (11) **2 295 908** (13) **A**

(43) Date of A Publication 12.06.1996

(21) Application No **9424856.4**

(22) Date of Filing **09.12.1994**

(71) Applicant(s)
GPT Limited

(Incorporated in the United Kingdom)

**PO Box 53, New Century Park, Telephone Road,
COVENTRY, CV3 1HJ, United Kingdom**

(72) Inventor(s)
Stephen William Marland

(74) Agent and/or Address for Service
**Henry Anthony Branfield
GEC Patent Department, Waterhouse Lane,
CHELMSFORD, Essex, CM1 2QX, United Kingdom**

(51) INT CL⁶
G06F 9/44

(52) UK CL (Edition O)
G4A AEF APX

(56) Documents Cited
GB 2231419 A

(58) Field of Search
UK CL (Edition N) **G4A AEF AFMP APX**
INT CL⁶ **G06F 9/44**
ONLINE : WPI

(54) **Updating EPROMs**

(57) Where a bootrom for a data processing system is stored in an EPROM it is advantageous to upgrade the bootrom in situ. However, if a catastrophic failure occurs there would be an incomplete or non-existent program. This may be avoided by using a method of updating a computer program contained within an electrically programmable read-only memory (EPROM), the EPROM having a sector containing a first computer program and a VARIABLE at a known fixed address. The method comprising the steps of:-

- (a) writing an upgraded program code into an empty sector of the EPROM and verifying it;
- (b) writing the upgraded program start address into a VECTOR address;
- (c) writing a specified value into the VARIABLE.

GB 2 295 908 A

Best Available Copy

EPROM UPDATES

Where a bootrom for a data processing system is stored in an EPROM, for example a Flash EPROM, it is advantageous for the bootrom to be subsequently upgraded in situ.

Typically this would be done by downloading a program to the system which would erase the EPROM and then reprogram it with the new code. However, this presents the problem that if a catastrophic failure occurs during the erase - reprogram cycle the EPROM would be left with an incomplete or non-existent program. This would mean that the system might in the limit have to be returned to the manufacturer or a service station for reprogramming or replacement of the bootrom since communication with the EPROM would be impossible.

According to the present invention there is provided a method of carrying out the upgrading of a computer program contained within an electrically programmable read-only memory (EPROM), the EPROM having a sector containing a first computer program and a VARIABLE at a known fixed address, comprising the steps of:-

- (a) writing an upgraded program code into an empty sector of the EPROM and verifying it;
- (b) writing the upgraded program start address into a VECTOR address;
- (c) writing a specified value into the VARIABLE.

The present invention will now be described by way of example and relates to a data

processing system which includes a bootprom which is stored in an electrically programmable read-only memory (EPROM). A bootprom is a computer program which is stored in a read-only memory (ROM), such as an EPROM, which provides the initial instructions set for the data processing system in which the ROM is installed. Such a procedure is known as "booting a computer".

Every time the bootprom is reset the program causes a VARIABLE at a known fixed address in the EPROM to be read. If the VARIABLE is not a specified value, then the bootprom continues as normal with the original computer program in the EPROM. If the VARIABLE is a specified value then a VECTOR is read at a known fixed address in the EPROM and the bootprom continues from an address in the EPROM given by the VECTOR and loads a new computer program from the EPROM.

If it becomes necessary to upgrade the computer program from the original program then the algorithm is as follows:-

1. Write the new program code into an empty sector in the EPROM and verify it.
2. Write the upgraded program start address into the VECTOR.
3. Write the specified value into the VARIABLE.

The bootprom is now totally enhanced. Code execution will begin using the original code but when the VARIABLE is checked and found to contain the specified variable, execution

will transfer to the new code as specified by the VECTOR.

If a catastrophic failure occurs at any point before the specified code word has been written to the VARIABLE in step 3, the system will revert to the original code.

Once the VARIABLE has been successfully written the procedure is complete and the system will use the NEW code.

If at some future time it is necessary to upgrade the code again the following algorithm is followed.

1. Modify the value in the VARIABLE by writing another value to it. This causes a reversion to the first version of code in the event that a reset should occur.
2. Erase the sectors containing the second version of code, the VARIABLE and the VECTOR.
3. Continue from step 1 in the first algorithm putting the new THIRD version of code into an empty (or erased) sector.

It is necessary that there is at least enough free space in the EPROM for the two versions of the program code AND the VARIABLE (2 bytes) and the VECTOR (typically 4 or 6 bytes for the Intel 386 family of microprocessors, for instance) simultaneously and that the new code has been fully proven before the upgrade. Also the VARIABLE and the VECTOR should not

be in a sector occupied by any of the original code. The specific value to be written to VARIABLE must be decided before the original bootprom program is written.

It would of course be possible where there is sufficient free space in the EPROM to put a third or further version of the code in the EPROM and not erase the second version of the code and simply modify the VECTOR to address the third or further version of the code.

The upgraded program may be a full program or a partial program which replaces a section of the original program code.

CLAIMS

1. A method of carrying out the upgrading of a computer program contained within an electrically programmable read-only memory (EPROM), the EPROM having a sector containing a first computer program and a VARIABLE and a VECTOR each at a known fixed address, comprising the steps of:-

- (a) writing an upgraded program code into an empty sector of the EPROM and verifying it;
- (b) writing the upgraded program start address into a VECTOR address;
- (c) writing a specified value into the VARIABLE.

2. A method as claimed in Claim 1, further comprising the steps of:

- (a) modifying the value in the VARIABLE;
- (b) writing a further upgraded program code into an empty sector of the EPROM and verifying it;
- (c) writing the further upgraded program start address into the VECTOR address;
- (d) rewriting the specified value into the variable.

3. A method as claimed in Claim 2, further comprising the step of erasing the upgraded program code before writing the further upgraded program code into the EPROM.

4. A method as claimed in Claim 1 and substantially as hereinbefore described.